Eurasian Journal of Analytical Chemistry
ISSN: 1306-3057 OPEN ACCESS 2017 12 (4): 246-258

Semantic Web Analysis Using Web Services

K. Anita Davamani, S. Amudha

Received: 12 December 2016 ▪ Revised: 15 January 2017 ▪ Accepted: 14 February 2017

Abstract: The Semantic Web is a major research initiative of the World Wide Web Consortium (W3C) to create a metadata-rich Web of resources that can describe themselves not only by how they should be displayed (HTML) or syntactically (XML), but also by the meaning of the metadata. The main intent of semantic web is to give machines better access to information resources so that they can be information intermediaries in support of humans. The idea is to build a network of content stored on the web and making it possible for machines to understand data and to satisfy requests from people and other machines. In order to carry out their tasks intelligent agents must communicate and understand meaning. The agent based method for semantic analysis enables computers to understand documents written in natural language. To realize the vision of semantic analysis we create markup of web services that makes them machine understandable and use-apparent. Also agent technology is developed that exploits this semantic markup to support automated web service composition and interoperability. Currently, a human must perform all the tasks in the web. With semantic markup of services, we can specify the information necessary for web service discovery as computer interpretable semantic markup at websites, and search engine can automatically locate appropriate services. The major semantic web services are automatic web service discovery, automatic web service execution, and automatic web service composition and interoperability.

Keywords: Semantic Web, Resources, Semantic Web Services, Metadata, Service Discovery.

INTRODUCTION

The Semantic Web is a major research initiative of the World Wide Web Consortium (W3C) to create a metadata-rich Web of resources that can describe themselves not only by how they should be displayed (HTML) or syntactically (XML), but also by the meaning of the metadata. It is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The idea is to build a network of content stored on the web – the Semantic Web – making it possible for machines to understand meaning of data and to satisfy requests from people and machines to use the web content. Semantics is the study of meaning in communication. Semantic Web is a group of methods and technologies to allow machines to understand the meaning - or "semantics" - of information on the World Wide Web. Its implementation requires adding semantic data. This allows machines process data based on semantic information so that computers can make inferences, understand resource descriptions and relations.

The purpose of semantic web is that Humans depend on web to carry out tasks. But web pages are designed to be read by humans not machines. So, computers cannot accomplish tasks without human direction. Semantic web is a vision of information that is understandable by computers, so computers can perform more of the tedious work involved in finding, combining, and acting upon information on the web. The main intent of the Semantic Web is to give machines much better access to information resources so they can be information intermediaries in support of humans.

The main idea of the new approach is that a software agent is assigned to each word of the text under consideration. Agents have access to a comprehensive repository of knowledge about possible meanings

K. Anita Davamani, Assistant Professor, Department of Computer Science and Engineering, BIST, BIHER, Bharath Institute of Higher Education & Research, Selaiyur, Chennai. E-mail: anitadavamani@gmail.com

S. Amudha, Assistant Professor, Department of Computer Science and Engineering, BIST, BIHER, Bharath Institute of Higher Education & Research, Selaiyur, Chennai. E-mail: amudha17s@gmail.com

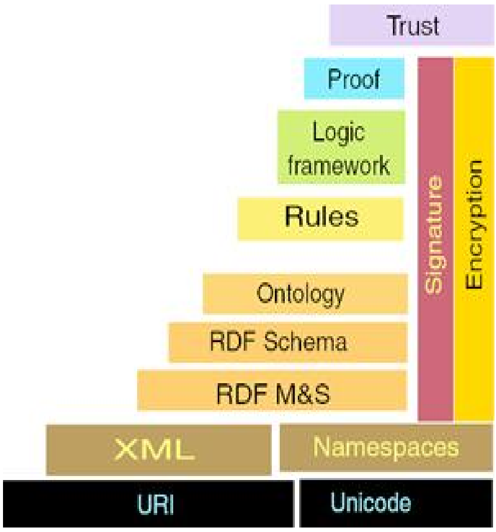
of words in the text and engage into negotiation with each other until a consensus is reached on meanings of each word and each sentence.

The agent-based method for defining semantics enables computers to understand contents of documents written in a natural language such as English. Possible applications of semantic analysis are numerous and include:

- Written communication between people and computers
- Written communication among computers
- Software translators
- Text referencing engines
- Semantic search engines
- Auto-abstracting engines
- Annotation and classification systems
- Semantic document-flow management systems

SEMANTIC WEB SEARCH

The Semantic Web layers are arranged following an increasing level of complexity from bottom to top. Higher layers functionality depends on lower ones. This design approach facilitates scalability and encourages using the simpler tools for the purpose at hand. All the layers are detailed in the next subsections.



A. URI and UNICODE

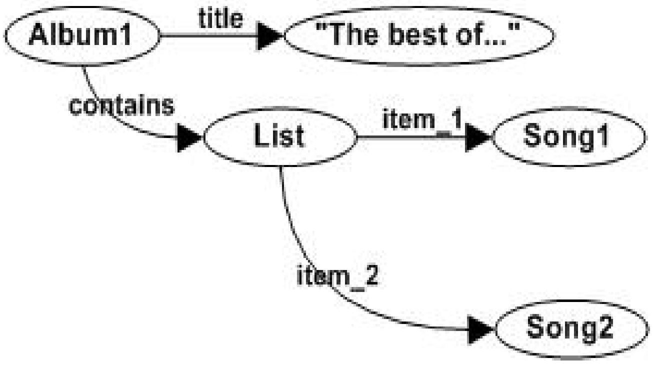
The two technologies that conform this layer are directly taken from the World Wide Web. URI provides global identifiers and UNICODE is a character-encoding standard that supports international characters. In few words, this layer provides the global perspective, already present in the WWW, for the Semantic Web.

B. XML and Namespaces

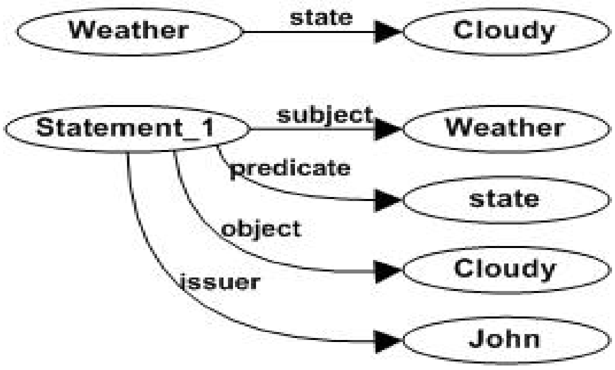
The Semantic Web should smoothly integrate with the Web. Therefore, it must be interwoven with Web documents. HTML is not enough to capture all that is going to be expressible in the Semantic Web. XML is a superset of HTML that can be used the serialization syntax for the Semantic Web. XML was initially tried but more recently other possibilities have been developed. They are presented and compared in the next section. Namespaces where added to XML to increase its modularization and the reuse of XML vocabularies in conjunction with XML Schemas. They are also used in the Semantic Web for the same purpose. Before XML, data was stored in flat file and database formats, where most data was proprietary to an application. XML came along and made data interoperable within a single domain, i.e., within the domain defined by a schema or a set of related schemas. By itself, XML provides syntactic interoperability only when both parties know and understand the element names used. If I label an element <price>12.00</price> and someone else labels it <cost>12.00<cost>, there‘s no way for a machine to know that those are the same thing without the aid of a separate, highly customized application to map between the elements. Semantic Web technologies help addresses this problem by making tags understandable not just to humans but to machines as well.

C. RDF Model and Syntax

The RDF Model and Syntax specification Becket04 defines the building blocks to realise the Semantic Web. This is the first layer that was specifically developed for it. This specification defines the RDF graph model and the RDF abstract syntax. The RDF graph model defines a structure composed of nodes and directed edges between nodes. The structure of nodes and edges conform directed graphs that model the network of terms and relations between terms of the Semantic Web. The nodes and relations are called resources and are identified by URIs. Each node has its own URI and there are different types of relations that also have an URI, they are called properties. Figure shows and **example of RDF graph model.**



Particular edges are identified by the triad composed by the origin node, the property and the destination node. Triads are called triples ore RDF statements and they are the RDF abstract syntax. Graphs can be serialized as a set of triples, one for each edge in the graph. Both representations are equivalent so the graph model can be reconstructed from the set of triples. Triples can also be assigned an explicit identifier, i.e. an URI. This process is called reification. A new node is created that represents the triple and it is associated to three nodes for the three triple components. The origin node is associated using the "subject" property, the property with the "predicate" property and the destination node with the "object" property. Reification is useful to say things about RDF statements.



Triple Reification Example

Abstract triples are the common model to which diverse data structures can be mapped. For instance, relational tables can be translated to a set of triples. Notwithstanding, triples are abstract entities. They are realized for communication using serialization syntaxes.

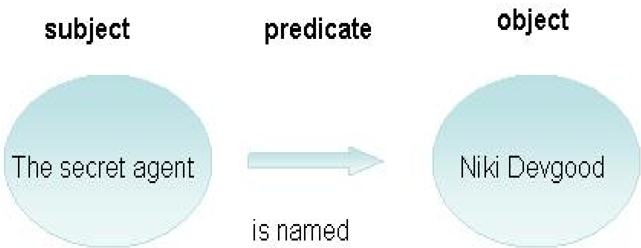
The XML syntax has already been introduced in the previous section; it facilitates integrating Semantic Web documents in the current HTML/XML web. The other possibilities are N-Triples and Notation 3 syntax, http://www.w3.org/DesignIssues/Notation3.html. The former is the nearest to the abstract form, a series of triples with subject, predicate and object identified by their URI. The latter uses many syntactic tricks to improve human readability and make serializations more compact. It is the more human aware syntax and, like XML serialization, it uses namespaces for modularization.

An official W3C recommendation, RDF is an XML-based standard for describing resources that exist on the Web, intranets, and extranets. RDF builds on existing XML and URI (Uniform Resource Identifier) technologies, using a URI to identify every resource, and using URIs to make statements about resources. RDF statements describe a resource (identified by a URI), the resource‗s properties, and the values of those properties. RDF statements are often referred to as ―triples‖ that consist of a subject, predicate, and object, which correspond to a resource (subject) a property (predicate), and a property value (object).

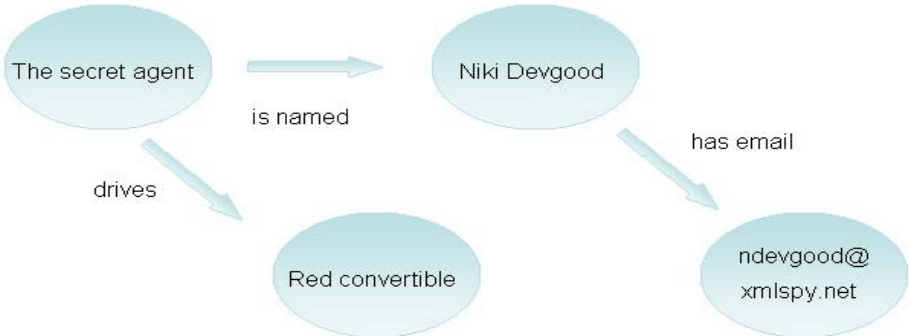
Table I

[resource]	[property]	[value]
The secret agent	Is	Niki Devgood
[subject]	[predicate]	[object]

RDF triples can be written with XML tags, and they are often conceptualized graphically as shown below:



After creating this triple, we can go on to create other triples to associate the agent with an email address, image, etc.



D. RDF Schema

Simple RDF provides the tools to construct semantic networks. They are a knowledge representation technology presented in the Semantic Networks section. Nonetheless, there is still a lack of many semantic network facilities not available with RDF. There are no defined taxonomical relations. They are defined in the RDF Schema specification [Brickley04]. Taxonomical relations leverage RDF to a knowledge representation language with capabilities similar to semantic networks. This enables taxonomical reasoning about the resources and the properties that relate them.

RDF Schema specification provides some primitives from semantic networks to define metadata vocabularies. RDF Schemas implement metadata vocabularies in a modular way, like XML Schemas. Schema primitives are also similar to Object Orientation constructs they also evolved from the semantic networks tradition. The more relevant ones are:

Type: it is a property that relates a resource to a Class to which it pertains. The resource is categorised as a member of this Class and thus it possesses its characteristics.

Class: it is a set of things that share some characteristics; they have a common conceptual abstraction. A class models the concepts present at the referential semantic level.

Sub Class Of: this property holds the taxonomical relations between classes. If class B is a subclass of class A, then class B has all the typical characteristics of class A plus some specific ones that can distinguish it from A.

E. Ontology

Ontologies are necessary when the expressiveness achieved with semantic network-like tools is not enough. Metadata vocabularies defined by RDF Schemas can be considered simplified ontologies. The tools included in this layer rise the developed vocabularies to the category of ontologies.

Ontologies, which were defined in the Knowledge Representation Ontology section, are specially suited to formalise domain specific knowledge. Once it is formalised, it can be easily interconnected with other formalisations. This facilitates the interoperability among independent communities and thus ontologies are one of the fundamental building blocks of the Semantic Web.

Description Logics are particularly suited for ontology creation. They were introduced in the corresponding Knowledge Representation subsection. The World Wide Web Consortium is currently developing a language for web ontologies, OWL. It is based on Description Logics and expressible in RDF so it integrates smoothly in the current Semantic Web initiative.

Description Logic makes possible to develop ontologies that are more expressible than RDF Schemas. Moreover, the particular computational properties of description logics reasoners make possible efficient classification and subsumption inferences.

F. Rules

The rules layer allows proof without full logic machinery. Similar rules are those used by the production systems presented in the Corresponding Knowledge Representation subsection. They capture dynamic knowledge as a set of conditions that must be fulfilled in order to achieve the set of consequences of the rule. The Semantic Web technology for this layer is the Semantic Web Rule Language (SWRL) [Horrocks04]. It is based on a previous initiative called Rule Modelling Language (RuleML) [Boley01]. As RuleML, SWRL covers the entire rule spectrum, from derivation and transformation rules to reaction rules. It can thus specify queries and inferences in Web ontologies, mappings between Web ontologies, and dynamic Web behaviours of workflows, services, and agents.

G. Logic

The purpose of this layer is to provide the features of FOL. First Order Logic was described as the most significant type of logic in the Logic types section. With FOL support, the Semantic Web has all the capabilities of logic available at a reasonable computation cost as shown in the Deduction section. There are some initiatives in this layer. One of the first alternatives was RDFLogic These extensions are supported by the CWM [Berners-Lee05] inference engine. Another more recent initiative is SWRL FOL [Patel-Schneider04], an extension of the rule language SWRL in order to cope with FOL features.

H. Proof

The use of inference engines in the Semantic Web makes it open, contrary to computer programs that apply the black-box principle. An inference engine can be asked why it has arrived to a conclusion, i.e. it gives proofs of their conclusions. There is also another important motivation for proofs. Inference engines problems are open questions that may require great or even infinite answer time. This is worse as the reasoning medium moves from simple taxonomical knowledge to full FOL. When possible, this problem can be reduced by providing reasoning engines pre-build demonstrations, proofs, that can be easily checked. Therefore, the idea is to write down the proofs when the problem is faced and it is easier to solve as the reasoning context is more constrained. Further, proofs are used whenever the problem is newly faced as a clue that facilitates reasoning on a wider content. Many inference engines specialised in particular subsets of logic have been presented so far.

I. Trust

This is the top layer of the Semantic Web architecture. Agents that want to work with the full-featured Semantic Web will be placed over it. They will conform the Web of Trust. The trust layer makes use of all the Semantic Web layers below. However, they do not provide the required functionality to trustily bind statements with their responsible parts. This is achieved with some additional technologies that are shown in the right part of the Semantic Web stack Figure.

The used tools are digital signature and encryption. Thus, the trust web will make intensive use of Public Key Infrastructures. They are already present in the Web, for instance as digital certificates identifying parties that sign digital contracts. Notwithstanding, there is not a widespread use of them. The premise is that Public Key Infrastructure is not of extended use because it is not a decentralised web structure. It is hierarchical and therefore rigid. What the Semantic Web might contribute here is a less constraining substrate of use. The web of trust is based on the graph structure of the Web. Moreover, it supports the dynamic construction of this graph. These features might enable the common use of Public Key Infrastructure in the future Web. To conclude, the final Semantic Web picture contains reasoning engines complemented with digital signatures to construct trust-engines. Then, a Trust Web can be developed with rules about which signed assertions are trusted depending on signer.

AGENT

Agent is one who acts for another, by authority from him; one entrusted with the business of another. A software agent is piece of software that acts for a user.

Agents are not strictly invoked for a task, but activate themselves. An Agent is a software object capable of contributing to the accomplishment of a task by

- Accessing domain knowledge
- Reasoning about it's task
- Composing meaningful messages
- Sending them to other agents or humans
- Interpreting received messages
- Making decisions based on domain knowledge and collected information
- Acting upon decisions in a meaningful manner

A *Multi-Agent System* is a software system consisting of agents competing or co-operating with each other with a view to accomplishing system tasks. The main principle of achieving goals within such system is a negotiation among agents, aimed at finding a balance between many different interests of individual agents. Ontology is a conceptual description of a domain of the Universe under consideration. Concepts are organized in terms of objects, processes, attributes and relations. Values defining instances of concepts are stored in associated databases. Concepts and values together form the domain knowledge. Syntactic Descriptor is a network of words linked by syntactic relations representing a grammatically correct sentence.

Semantic Descriptor is a network of grammatically and semantically compatible words, which represents a computer readable interpretation of the meaning of a text. If semantic ontology describes all possible meanings of words in a domain, a semantic descriptor describes the meaning of a particular text. Self-organization is the capability of a system to autonomously, ie, without human intervention, modify existing and/or establish new relationships among its components with a view to increasing a given value or recovering from a disturbance, such as, an unexpected addition or subtraction of a component. In the context of text understanding any autonomous change of a link between two agents representing different meanings of words is considered as a step in the process of self-organization. Evolution is the capability of a system to autonomously modify its components and/or links in response, or in anticipation of changes in its environment.

In the context of text understanding any autonomous update of Ontology based on the newly acquired information is considered as a step in the process of evolution.

AGENT BASED SEMANTIC WEB

In order to carry out their required tasks, intelligent agents must communicate and understand meaning. They must advertise their capabilities, and recognize the capabilities of other agents. They must locate meaningful information resources on the Web and combine them in meaningful ways to perform tasks. They need to recognize, interpret, and respond to communication acts from other agents. As mentioned before, when agents communicate with each other, there needs to be some way to ensure that the meaning of what one agent "says" is accurately conveyed to the other agent. The simplest and most common approach is to build-in the semantics. That is, just assume that all agents are using the same terms to mean the same things.

The assumption could be implicit and informal, or it could be an explicit agreement among all parties to commit to using the same terms in a pre-defined manner, i.e. a standard. This only works, however, when one has full control over what agents exist and what they might communicate. In reality, agents need to interact in a much wider world, where it cannot be assumed that other agents will use the same terms, or if they do, it cannot be assumed that the terms will mean the same thing. Therefore, we need a way for an agent to discover what another agent means when it communicates. In order for this to happen, agents will need to publicly declare exactly what terms it is using and what they mean. This specification is commonly referred to as the agent's ontology. There is a great interdependence of agent technology and ontologies.

AGENT BASED METHOD FOR SEMANTIC ANALYSIS

The method consists of the following four steps:

- Morphological analysis
- Syntactic analysis
- Semantic analysis
- Pragmatics

The text is divided into sentences. Sentences are fed into the meaning extraction process one by one.

A. Proof

- An agent is assigned to each word in the sentence
- Word Agents access Ontology and acquire relevant knowledge on morphology
- Word Agents execute morphological analysis of the sentence and establish characteristics of each word, such as gender, number, case, time, etc.
- If morphological analysis results in polysemy, ie, a situation in which some words could play several roles in a sentence (a noun or adjective or verb), several agents are assigned to the same word each representing one of its possible roles

B. Syntactical Analysis

- Word Agents access Ontology and acquire relevant knowledge on syntax.
- Word Agents execute syntactical analysis where they aim at identifying the syntactical structure of the sentence. For example, a Subject searches for a Predicate of the same gender and number, and a Predicate looks for a suitable Subject and Objects. Conflicts are resolved through a process of negotiation. A grammatically correct sentence is represented by means of a Syntactic Descriptor. If results of the syntactical analysis are ambiguous, ie, several variants of the syntactic structure of the sentence under consideration are feasible, each feasible variant is represented by a different Syntactic Descriptor.

C. Semantic Analysis

- Word Agents access Ontology and acquire relevant knowledge on semantics.
- Each grammatically correct version of the sentence under consideration is subjected to semantic analysis. This analysis is aimed at establishing the semantic compatibility of words in each grammatically correct sentence. Word Agents learn from Ontology possible meanings of words that they represent and by consulting each other attempt to eliminate inappropriate alternatives
- Once agents agree on a grammatically and semantically correct sentence, they create a Semantic Descriptor of the sentence, which is a network of concepts and values contained in the sentence.
- If a solution that satisfies all agents cannot be found, agents compose a message to the user explaining the difficulties and suggesting how the issues could be resolved.
- Each new grammatically and semantically correct sentence generated by the steps 1 - 11 is checked for semantic compatibility with Semantic Descriptors of preceding sentences. In the process agents may decide to modify previously agreed semantic interpretations of words or sentences (self-organisation).
- When all sentences are processed, the final Semantic Descriptor of the whole document is constructed thus providing a computer readable semantic interpretation of the text.

D. Pragmatics

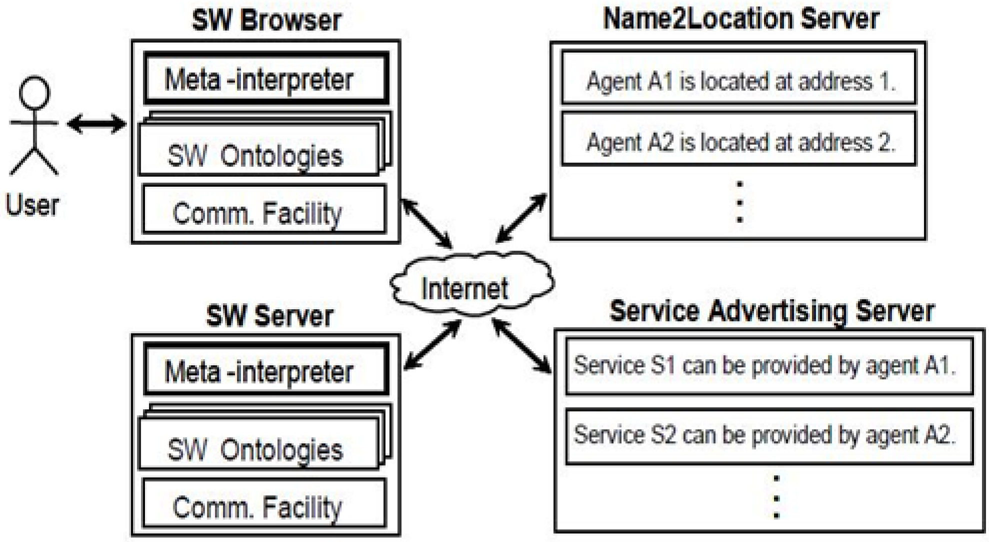
- Word Agents access Ontology and acquire relevant knowledge on pragmatics, which is closely related to the application at hand.
- At this stage agents consider their application-oriented tasks and decide if they need to execute any additional processes. For example, if the application is a Person – Computer Dialog, agents may decide that they need to ask the user to supply some additional information; if the application is a Search Engine, agents will compare the Semantic Descriptor of the search request with Semantic Descriptors of available search results. If the application is a Classifier, agents will compare Semantic Descriptors of different documents and form groups of documents with semantic proximity.

Main features of the proposed method.

- Decision making rules are specified in ontology, which incorporates general knowledge on text understanding, language-oriented rules and specific knowledge on the problem domain.
- Every word in the text under consideration is given the opportunity to autonomously and pro-actively search for its own meaning using knowledge available in ontology.
- Tentative decisions are reached through a process of consultation and negotiation among all Word Agents.
- The final decision on the meaning of every word is reached through a consensus among all Word Agents.
- Semantic Descriptors are produced for individual sentences and for the whole text.
- The extraction of meanings follows an autonomous trial-and-error pattern (self organisation).

The process of meaning extraction can be regulated by modifying ontology.

FRAMEWORK



The meta-logical system for one agent consists of three main parts: meta-programs for multiple ontologies, a meta-interpreter, and the communication facility. Each meta-program contains meta-logical representations of ontologies obtained from the transformation of these ontologies defined in RDF, RDFS, and OWL. Some elements in one ontology may be related to some elements in another. The meta-interprete is the inference engine for infering implicit information from the multiple ontologies. The communication facility supports the communication among the agents.

A. How Query is Processed

When several agents of this kind are formed as a community, the way the multi-agent system works is that initially the user queries an SW browser to get answers from an SW ontology on SW. The browser can perform two alternative ways. Firstly, it may retrieve this ontology from SW, transform it into a metaprogram, and then reason with the program to infer the answers; if some elements in this ontology are related to some elements of another ontology, the interpreter will try to reason with that ontology in itself (by retrieving it first), or request reasoning of that ontology in an SW server and obtain the answers from that server, and this scenario may repeat itself. For the browser to be able to retrieve an ontology, it must know which server the ontology belongs to, and how and where to access to it. This is the ontology‘s meta-information provided in the ontology. The browser will use this to contact with that server and request that ontology from it, or to pass a query to that server so that the server can derive an answer from the ontology.

Alternatively, the browser passes the query to an SW server to answer and gets the answers back for the user. The server infers those answers based on its inferential results which sometimes also require support of the inferential results derived from other servers. In case the browser does not know which server can answer that query, it will consult the Service Advertising Server which gathers information telling which server can provide what service. The browser then uses this information to communicate with the selected server directly.

For the browser to communicate to any server as said earlier, having known the server name the browser will pass the name to the Name2Location server to obtain the server location and then make contact with that server at that location. Note that conceptually the term ‗location ‘we use here is intended to be an abstract one; an agent location could be the place, such as an address (IP address) on the Internet, or even a (postal) address.

SEMANTIC WEB SERVICES

To realize our vision of Semantic Web services we are creating semantic markup of Web services that makes them machine understandable and use-apparent. We are also developing agent technology that exploits this semantic markup to support automated Web service composition and interoperability. Driving\ the development of our markup and agent technology are the automation tasks that semantic markup of Web services will enable —in particular, service discovery, execution, and composition and interoperation.

A. Automatic Web Service Discovery

It involves automatically locating Web services that provide a particular service and that adhere to requested properties. A user might say, for example, ―Find a service that sells airline tickets between San Francisco and Toronto and that accepts payment by Diner‗s Club credit card.‖ Currently, a human must perform this task, first using a search engine to find a service and then either reading the Web page associated with that service or executing the service to see whether it adheres to the requested properties. With semantic markup of services, we can specify the information necessary for Web service discovery as computer-interpretable semantic markup at the service Web sites, and a service registry or (ontology-enhanced) search engine can automatically locate appropriate services.

B. Automatic Web Service Execution

It involves a computer program or agent automatically executing an identified Web service. A user could request, ―Buy me an airline ticket from www.acmetravel.com on UAL Flight 1234 from San Francisco to Toronto on 3 March.‖ To execute a particular service on today‗s Web, such as buying an airline ticket, a user generally must go to the Web site offering that service, fill out a form, and click a button to execute the service. Alternately, the user might send an http request directly to the service URL with the appropriate parameters encoded. Either case requires a human to understand what information is required to execute the service and to interpret the information the service returns. Semantic markup of Web services provides a declarative, computer-interpretable API for executing services. The markup tells the agent what input is necessary, what information will be returned, and how to execute—and potentially interact with—the service automatically.

C. Automatic web Service Composition and Interoperation

It involves the automatic selection, composition, and interoperation of appropriate Web services to perform some task, given a high-level description of the task‗s objective. A user might say, ―Make the travel arrangements for my IJCAI 2001 conference trip. Currently, if some task requires a composition of Web services that must interoperate, then the user must select the Web services, manually specify the composition, ensure that any software for interoperation is custom-created, and provide the input at choice points (for example, selecting a flight from among several options). With semantic markup of Web services, the information necessary to select, compose, and respond to service is encoded at the service Web sites. We can write software to manipulate this markup, together with a specification of the task‗s objectives, to achieve the task automatically. Service composition and interoperation leverage automatic discovery and execution. Of these three tasks, none is entirely realizable with today‗s Web, primarily because of a lack of content markup and a suitable markup language. Academic research on Web service discovery is growing out of agent matchmaking research such as the Lark system, 6 which proposes a representation for annotating agent capabilities so that they can be located and brokered. Recent industrial efforts have focused primarily on improving Web service discovery and aspects of service execution through initiatives such as the Universal Description, Discovery, and Integration (UDDI) standard service registry; the XML based Web Service Description Language (WSDL), released in September 2000 as a framework-independent Web service description language; and ebXML, an initiative of the United Nations and OASIS (Organization for the Advancement of Structured Information Standards) to standardize a framework for trading partner interchange.

CASE STUDY: WOLFRAM ALPHA

Wolfram Alpha (styled Wolfram |Alpha) is an answer engine developed by Wolfram Research. It is an online service that answers factual queries directly by computing the answer from structured data, rather than providing a list of documents or web pages that might contain the answer as a search engine would. It was announced in March 2009 by Stephen Wolfram, and was released to the public on May 15, 2009. It was voted the greatest computer innovation of 2009 by Popular Science.

A. Design

Users submit queries and computation requests via a text field. Wolfram Alpha then computes and provides answers and relevant visualizations from a core knowledge base of curated, structured data. Wolfram Alpha thus differs from semantic search engines, which index a large number of answers and then try to match the question to one. In this way it has many parallels with Cyc, a project aimed since the 1980s at developing a common-sense inference engine. Wolfram Alpha is built on Wolfram's earlier flagship product, Mathematica, a complete functional-programming package which encompasses computer algebra, symbolic and numerical computation, visualization, and statistics capabilities. With Mathematica running in the background, it is suited to answer mathematical questions.

The answer usually presents a human-readable solution. Alpha also incorporates elements of web Mathematica in delivering its content. Wolfram Alpha is also capable of responding to increasingly complex, natural-language fact-based questions such as:

"Where was Mary Robinson born?"
"How old was Queen Elizabeth II in 1974?"
What is the forty-eighth smallest country by GDP per capita?" yields Senegal, $1090 per year.

Also, one can input the name of a website, and it will return relevant information about the site, including its hosting location, site rank, number of visitors and more.

The database currently includes hundreds of datasets, including current and historical weather, drug data, star charts, currency conversion, and many others. The datasets have been accumulated over approximately two years, and are expected to continue to grow. The range of questions that can be answered is also expected to grow with the expansion of the datasets.

B. Technology

Wolfram Alpha is written in about five million lines of Mathematica (using web Mathematica and grid Mathematica) code and runs on 10,000 CPUs (though the number was upgraded for the launch). As well as being a web site, Wolfram Alpha provides an API that delivers computational answers to other applications. One such application is the Bing search engine.8.3 System requirements. Wolfram Alpha requires an up-to-date web browser. On older browsers, it does not display text correctly and renders a message that states Sorry... Wolfram |Alpha requires a more up-to-date web browser... on its home page. An iPhone application was released on 19 October 2009 to provide an optimized interface for mobile access, priced at $49.99; in March 2010, the price was cut to $1.99 and made an iPad / iPhone universal application (users who purchased at the higher price were refunded the balance). The Wolfram Alpha website is also supported by the Opera Mini mobile browser. Wolfram Alpha launched a version of its Web site in app form on Wednesday 6th October 2010, which costs $1.99 to download from the Android Market.

C. How Query is Processed

Wolfram |Alpha determine the real meaning of the question asked. Natural language parsers attempt to determine the domain(s) of the question. If Wolfram |Alpha doesn't understand meaning behind some general question, it doesn't return any pages. But Google will give you several web pages. It also prompts you to disambiguate queries that apply to multiple domains. It generates output by doing computations from its own internal knowledge base.

CONCLUSION

The integration of agent technology and ontologies has made significant impact on the use of web services .This gives the ability to extend programs to more efficiently perform tasks for users with less human intervention. Unifying these research areas and bringing to fruition a web teeming with complex, "intelligent" agents is both possible and practical. Although a number of research challenges still remain. The pieces are coming together, and thus, the semantic web of agents is no longer a science fiction future. It is a practical application on which to focus current efforts.

REFERENCES

[1] Das, J., Das, M. P., & Velusamy, P. (2013). Sesbania grandiflora leaf extract mediated green synthesis of antibacterial silver nanoparticles against selected human pathogens. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, *104*, 265-270.

[2] Umanath, K.P.S.S.K., Palanikumar, K., & Selvamani, S. T. (2013). Analysis of dry sliding wear behaviour of Al6061/SiC/Al2O3 hybrid metal matrix composites. *Composites Part B: Engineering*, *53*, 159-168.

[3] Udayakumar, R., Khanaa, V., Saravanan, T., & Saritha, G. (1786). Cross layer optimization for wireless network (WIMAX). *Middle-East Journal of Scientific Research*, *16*(12), 1786-1789.

[4] Kumaravel, A., & Rangarajan, K. (2013). Algorithm for automaton specification for exploring dynamic labyrinths. *Indian Journal of Science and Technology*, *6*(5S), 4554-4559.

[5] Pieger, S., Salman, A., & Bidra, A.S. (2014). Clinical outcomes of lithium disilicate single crowns and partial fixed dental prostheses: a systematic review. *The Journal of prosthetic dentistry*, *112*(1), 22-30.

[6] Vijayaraghavan, K., Nalini, S.K., Prakash, N.U., & Madhankumar, D. (2012). One step green synthesis of silver nano/microparticles using extracts of Trachyspermum ammi and Papaver somniferum. *Colloids and Surfaces B: Biointerfaces*, *94*, 114-117.

[7] Khanaa, V., Mohanta, K., & Satheesh, B. (2013). Comparative study of uwb communications over fiber using direct and external modulations. *Indian Journal of Science and Technology*, *6*(6), 4845-4847.

[8] Khanaa, V., Thooyamani, K. P., & Udayakumar, R. (1798). Cognitive radio based network for ISM band real time embedded system. *Middle-East Journal of Scientific Research*, *16*(12), 1798-1800.

[9] Vijayaraghavan, K., Nalini, S.K., Prakash, N.U., & Madhankumar, D. (2012). Biomimetic synthesis of silver nanoparticles by aqueous extract of Syzygium aromaticum. *Materials Letters*, *75*, 33-35

[10] Caroline, M.L., Sankar, R., Indirani, R.M., & Vasudevan, S. (2009). Growth, optical, thermal and dielectric studies of an amino acid organic nonlinear optical material: l-Alanine. *Materials Chemistry and Physics*, *114*(1), 490-494.

[11] Kumaravel, A., & Pradeepa, R. (2013). Efficient molecule reduction for drug design by intelligent search methods. *International Journal of Pharma and Bio Sciences*, *4*(2), B1023-B1029.

[12] Kaviyarasu, K., Manikandan, E., Kennedy, J., Jayachandran, M., Ladchumananandasiivam, R., De Gomes, U. U., & Maaza, M. (2016). Synthesis and characterization studies of NiO nanorods for enhancing solar cell efficiency using photon upconversion materials. *Ceramics International*, *42*(7), 8385-8394.

[13] Sengottuvel, P., Satishkumar, S., & Dinakaran, D. (2013). Optimization of multiple characteristics of EDM parameters based on desirability approach and fuzzy modeling. *Procedia Engineering*, *64*, 1069-1078.

[14] Anbuselvi S., Chellaram, C., Jonesh S., Jayanthi L., & Edward J.K.P. (2009). Bioactive potential of coral associated gastropod, Trochus tentorium of Gulf of Mannar, Southeastern India. *J. Med. Sci,* 9(5), 240-244.

[15] Kaviyarasu, K., Ayeshamariam, A., Manikandan, E., Kennedy, J., Ladchumananandasivam, R., Gomes, U. U., & Maaza, M. (2016). Solution processing of CuSe quantum dots: Photocatalytic activity under RhB for UV and visible-light solar irradiation. *Materials Science and Engineering: B*, *210*, 1-9.

[16] Kumaravel, A., & Udayakumar, R. (2013). Web portal visits patterns predicted by intuitionistic fuzzy approach. *Indian Journal of Science and Technology*, *6*(5S), 4549-4553.

[17] Srinivasan, V., & Saravanan, T. (2013). Reformation and market design of power sector. *Middle-East Journal of Scientific Research*, *16*(12), 1763-1767.

[18] Kaviyarasu, K., Manikandan, E., Kennedy, J., & Maaza, M. (2015). A comparative study on the morphological features of highly ordered MgO: AgO nanocube arrays prepared via a hydrothermal method. *RSC Advances*, *5*(100), 82421-82428.

[19] Kumaravel, A., & Udhayakumarapandian, D. (2013). Consruction of meta classifiers for apple scab infections. *International Journal of Pharma and Bio Sciences*, *4*(4), B1207-B1213.

[20] Sankari, S. L., Masthan, K. M. K., Babu, N. A., Bhattacharjee, T., & Elumalai, M. (2012). Apoptosis in cancer-an update. *Asian Pacific journal of cancer prevention*, *13*(10), 4873-4878

[21] Harish, B. N., & Menezes, G. A. (2011). Antimicrobial resistance in typhoidal salmonellae. *Indian journal of medical microbiology*, *29*(3), 223-229.

[22] Manikandan, A., Manikandan, E., Meenatchi, B., Vadivel, S., Jaganathan, S. K., Ladchumananandasivam, R., & Aanand, J. S. (2017). Rare earth element (REE) lanthanum doped zinc oxide (La: ZnO) nanomaterials: synthesis structural optical and antibacterial studies. *Journal of Alloys and Compounds*, *723*, 1155-1161.

[23] Caroline, M. L., & Vasudevan, S. (2008). Growth and characterization of an organic nonlinear optical material: L-alanine alaninium nitrate. *Materials Letters*, *62*(15), 2245-2248.

[24] Saravanan T., Srinivasan V., Udayakumar R. (2013). A approach for visualization of atherosclerosis in coronary artery, Middle - East Journal of Scientific Research, 18(12), 1713-1717.

[25] Poongothai, S., Ilavarasan, R., & Karrunakaran, C.M. (2010). Simultaneous and accurate determination of vitamins B1, B6, B12 and alpha-lipoic acid in multivitamin capsule by reverse-phase high performance liquid chromatographic method. *International Journal of Pharmacy and Pharmaceutical Sciences*, *2*(4), 133-139.

[26] Udayakumar, R., Khanaa, V., & Saravanan, T. (2013). Synthesis and structural characterization of thin films of SnO 2 prepared by spray pyrolysis technique. *Indian Journal of Science and Technology*, *6*(6), 4754-4757

[27] Anbazhagan, R., Satheesh, B., & Gopalakrishnan, K. (2013). Mathematical modeling and simulation of modern cars in the role of stability analysis. *Indian Journal of Science and Technology*, *6*(5S), 4633-4641.

[28] Caroline, M.L., & Vasudevan, S. (2009). Growth and characterization of bis thiourea cadmium iodide: A semiorganic single crystal. *Materials Chemistry and Physics*, *113*(2-3), 670-674.

[29] Sharmila, S., Jeyanthi Rebecca, L., & Das, M. P. (2012). Production of Biodiesel from Chaetomorpha antennina and Gracilaria corticata. *Journal of Chemical and Pharmaceutical Research*, *4*(11), 4870-4874.

[30] Thooyamani, K.P., Khanaa, V., & Udayakumar, R. (2013). An integrated agent system for e-mail coordination using jade. *Indian Journal of Science and Technology*, *6*(6), 4758-4761.

[31] Caroline, M. L., Kandasamy, A., Mohan, R., & Vasudevan, S. (2009). Growth and characterization of dichlorobis l-proline Zn (II): A semiorganic nonlinear optical single crystal. *Journal of Crystal Growth*, *311*(4), 1161-1165.

[32] Caroline, M.L., & Vasudevan, S. (2009). Growth and characterization of L-phenylalanine nitric acid, a new organic nonlinear optical material. *Materials Letters*, *63*(1), 41-44.

[33] Kaviyarasu, K., Xolile Fuku, Genene T. Mola, E. Manikandan, J. Kennedy, and M. Maaza. Photoluminescence of well-aligned ZnO doped CeO2 nanoplatelets by a solvothermal route. *Materials Letters*, *183*(2016), 351-354.

[34] Saravanan, T., & Saritha, G. (2013). Buck converter with a variable number of predictive current distributing method. *Indian Journal of Science and Technology*, *6*(5S), 4583-4588.

[35] Parthasarathy, R., Ilavarasan, R., & Karrunakaran, C. M. (2009). Antidiabetic activity of Thespesia Populnea bark and leaf extract against streptozotocin induced diabetic rats. *International Journal of PharmTech Research*, *1*(4), 1069-1072.

[36] Hanirex, D. K., & Kaliyamurthie, K. P. (2013). Multi-classification approach for detecting thyroid attacks. *International Journal of Pharma and Bio Sciences,* 4(3), B1246-B1251

[37] Kandasamy, A., Mohan, R., Lydia Caroline, M., & Vasudevan, S. (2008). Nucleation kinetics, growth, solubility and dielectric studies of L-proline cadmium chloride monohydrate semi organic nonlinear optical single crystal. *Crystal Research and Technology: Journal of Experimental and Industrial Crystallography*, *43*(2), 186-192.

[38] Srinivasan, V., Saravanan, T., Udayakumar, R., & Saritha, G. (2013). Specific absorption rate in the cell phone user's head. *Middle-East Journal of Scientific Research*, *16*(12), 1748-50.

[39] Udayakumar R., Khanaa V., & Saravanan T. (2013). Chromatic dispersion compensation in optical fiber communication system and its simulation, Indian Journal of Science and Technology, 6(6), 4762-4766.

[40] Vijayaragavan, S.P., Karthik, B., Kiran, T.V.U., & Sundar Raj, M. (1990). Robotic surveillance for patient care in hospitals. *Middle-East Journal of Scientific Research*, *16*(12), 1820-1824.

[41] Mohamed Ibrahim, B., & Dr. Mohamed Shanavas, A.R. (2015).An Approach to Predict SOA Security Vulnerabilities using Feed Forward Artificial Neural Networks. *The SIJ Transactions on Computer Networks & Communication Engineering (CNCE), 3*(3), 1-5.

[42] Yen, M.H., Lin, Y.H., Chang, Y.C., & Tsai, P. J. (2015).The Implementation of 8051 MCU for IC-EMC Testing. *The SIJ Transactions on Computer Networks & Communication Engineering (CNCE), 3*(5), 1-6.

[43] Mohammed, M., & Abdessadek, A. (2016).Weight Distribution and Bounds of Turbo-Code with 3 Dimensions. *The SIJ Transactions on Computer Networks & Communication Engineering (CNCE), 4*(2), 7-12.

[44] Dr. Malhotra, R., & Sachdeva, B. (2016).Multilingual Evaluation of the DSR, DSDV and AODV Routing Protocols in Mobile Ad Hoc Networks. *The SIJ Transactions on Computer Networks & Communication Engineering (CNCE), 4*(3), 7-13.

[45] Prathibha, P.H., & Dr.Chandran, C.P. (2016). Classification Mining SNPs from Leukaemia Cancer Dataset Using Linear Classifier with ACO. *Bonfring International Journal of Data Mining, 6*(2), 10-15.

[46] Sadeghi, K., & Hashemi, S.S. (2015). Customer selection to pay more to browser on credit card of using data mining and data warehouse. *International Academic Journal of Innovative Research, 2*(4), 28-34.

[47] Sundhar, C., & Archana, D. (2014). Automatic Screening of Fundus Images for Detection of Diabetic Retinopathy. *International Journal of Communication and Computer Technologies, 2*(1), 29-35.

[48] Elijah, and Dilber, M.N. (2017). Complete Analysis of Fault Tolerance Schemes in Mobile Agents for a Reliable Mobile Agent Computation. *Bonfring International Journal of Industrial Engineering and Management Science, 7*(1), 20-24.

[49] Toupchi, M., & Abolghasempur, S.A. (2015). Modify improved ant colony for fuzzy Clustering in image segmentation. *International Academic Journal of Science and Engineering, 2*(4), 19-28.

[50] Agnes Christy, V., & Navaneetha Velammal, M. (2014). Analysis and Design of Low Power Dynamic Memory using FVD and SPD Methods. *International Journal of System Design and Information Processing, 2*(2), 40-44.